

Глава 1

Класс $PSPACE$.

1.1 Класс $PSPACE$ и игры с полиномиальным числом ходов.

Определение 1.1 Класс $PSPACE$ состоит из всех языков $L \subset \Sigma^*$, распознаваемых на машинах Тьюринга с полиномиальным ограничением на зону вычисления, т.е. для которых существует машина Тьюринга M , вычисляющая значения предиката “ $v \in L$ ” на словах $v \in \Sigma^*$, причем

$$S_M(n) = \max_{|v|=n} S_M(v)$$

есть функция полиномиального роста.

Это определение полностью аналогично определению класса P , поэтому к нему также следует отнести все соответствующие комментарии из лекции ???. В то же время класс $PSPACE$ гораздо богаче класса P и, по-видимому, уже содержит в себе большинство практически значимых вычислительных задач.

Класс $PSPACE$ содержит в себе все рассмотренные ранее сложностные классы (P , NP , BPP , PH) кроме “неравномерного” класса $P/Poly$. Это вытекает из следующего игрового описания класса $PSPACE$.

Оказывается, что все языки из $PSPACE$ и только они могут быть описаны конечными играми, в которых число ходов не фиксировано, а задается полиномом от длины начальной конфигурации. Такая игра описывается двумя полиномами p , q и предикатом выигрыша $R(x, \bar{w})$, принадлежащим классу P . Как и раньше, $x \in \{0, 1\}^*$ – начальная конфигурация, а $\bar{w} = w_1 b_1 w_2 b_2 \dots \in \{0, 1\}^*$ – слово, составленное из чередующихся ходов двух игроков (M) и (A). Для данного x длины ходов

одинаковы, $|w_i| = |b_i| = p(|x|)$, и количество ходов $q(|x|)$ – тоже. (Хотя, как обсуждалось раньше, эти условия можно ослабить без изменения класса распознаваемых языков до $|w_i| = |b_i| < p(|x|)$ и количества ходов $< q(|x|)$.) Для определенности будем считать, что первый ход всегда делает (М). Язык $L \subset \{0, 1\}^*$ распознается игрой, если он состоит из всех начальных конфигураций, для которых у (М) есть выигрышная стратегия, т.е.

$$x \in L \Leftrightarrow \underbrace{\exists^p w_1 \forall^p b_1 \dots}_{q(|x|)} R(x, w_1 b_1 \dots).$$

Легко видеть, что игры с полиномиальным числом ходов моделируют ограниченные игры (из определения PH) – им соответствуют предикаты выигрыша, существенно зависящие только от фиксированного числа начальных ходов. Поэтому все языки из PH распознаются и такими играми. Ниже термин игра будет означать именно игры с полиномиальным числом шагов (их также называют *детерминированными интерактивными протоколами*).

1.2 Моделирование игры.

Пусть даны игра (p, q, R) и начальная конфигурация x . Рассмотрим *дерево игры* (см. Рис.??). Оно состоит из всевозможных $q(|x|)$ -элементных последовательностей ходов $w_1 b_1 w_2 b_2 \dots$ (ходы соответствуют ребрам; ребра ориентированы от корня к листьям). Каждый путь \bar{w} из корня к листьям задает некоторую партию. На соответствующем листе запишем ее результат, т.е. значение $R(x, \bar{w})$.

Как выяснить существование выигрышной стратегии для (М):

Все ребра-ходы (М) назовем \vee -ребрами, а все остальные – \wedge -ребрами. В вершинах, из которой выходят \vee -ребра, поместим функциональный элемент OR , а в вершинах, из которой выходят \wedge -ребра, поместим AND . Получим схему из функциональных элементов AND , OR , арности $2^{p(|x|)}$ каждый. Пометки на листьях (значения предиката R) будем считать входными и продолжим эту разметку на все вершины дерева согласно схеме. Если в корне получится 1, то у (М) есть выигрышная стратегия; если 0 – нет.

Как (М) должен играть, если в корне стоит 1 : Он каждый раз должен выбирать то ребро, которое ведет в вершину с 1.

Прямая реализация этого метода распознавания языков приводит к экспоненциальным затратам памяти на хранение дерева игры. Но на самом деле можно обойтись и полиномиальной памятью, реализовав вычисление булевых пометок обходом дерева "влево-вниз".

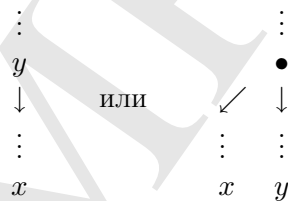
1.3 Моделирование на полиномиальной памяти.

Теорема 1.2 Если язык L распознается игрой с полиномиальным количеством ходов, то $L \in PSPACE$.

Доказательство. Достаточно показать, что вычисление значения построенной выше $\{AND, OR\}$ -схемы можно реализовать на полиномиальной зоне, не распределяя память под всю схему целиком.

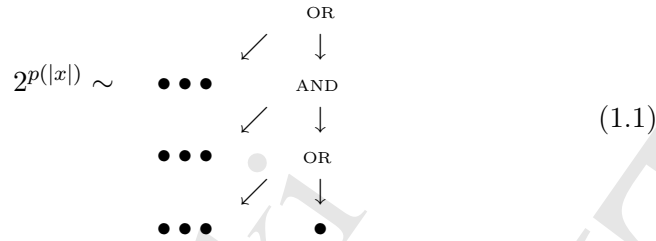
Итак, схема представляет собой корневое дерево высоты $q(|x|)$ с порядком ветвления $2^{p(|x|)}$. Вершины одного яруса – одинаковые функциональные элементы (либо все – AND , либо все – OR), причем AND и OR ярусы чередуются, а в корне стоит OR . Входные значения (на листьях) вычисляются на полиномиальной зоне с помощью предиката R , если известен путь из корня к листу. Предполагаем, что ребра, выходящие из одной вершины, упорядочены (слева направо).

На множестве всех вершин зададим линейный порядок "левее или ниже": $x < y$ если путь из корня в x есть собственное продолжение пути из корня в y или есть собственное ответвление от этого пути влево:

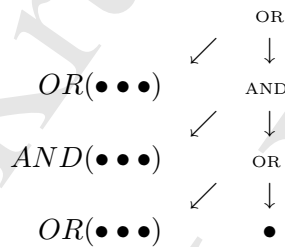


Предлагается последовательно вычислять значения схемы в вершинах дерева в этом порядке (от левой нижней вершины к корню). При этом для вычисления значений в вершинах $z > x$, достаточно знать значения в вершине x и всех вершинах $y < x$, расположенных непосредственно

ниже одной из вершин на пути из корня к x :



Размер подграфа (1.1) остается экспоненциальным за счет групп вершин, обозначенных тремя жирными точками. Но каждая такая группа расположена на одном ярусе, поэтому в значения схемы во всех вершинах $z > x$ войдет $AND(\bullet \bullet \bullet)$ или $OR(\bullet \bullet \bullet)$ в зависимости от четности номера яруса. Таким образом, в каждый момент времени достаточно хранить подграф полиномиального размера:



■

1.4 Игровая характеристика класса $PSPACE$.

Теорема 1.3 *Каждый язык $L \in PSPACE$ распознается некоторой игрой с полиномиальным числом ходов.*

Доказательство. Пусть машина Тьюринга M распознает язык L на зоне, ограниченной полиномом p . По теореме ?? ее время работы можно оценить сверху функцией $2^{q(n)}$, где q некоторый полином. Для данного входного слова x рассмотрим последовательность мгновенных конфигураций $C(t)$, $t = 0, \dots, 2^{q(|x|)}$, составляющих вычисление M на входе x . $C(t)$ есть содержимое лент, положение головок и состояние машины после t шагов вычисления; если к моменту t вычисление уже закончилось, то полагаем $C(t) := C(t - 1)$. По входу x однозначно восстанавливается начальная конфигурация $C(0)$. Будем предполагать, что заключительная конфигурация с ответом “да” также стандартизована (рабочие ленты

очищены, на выходной — ответ 1, головки — в стандартных позициях), так что ее вид можно восстановить по x не проводя всех вычислений. Пусть $next(C)$ обозначает следующую за C конфигурацию.

Игра для распознавания принадлежности $x \in L$. Игрок (М) старается убедить (А), что он обладает правильной последовательностью конфигураций вычисления $M(x)$ и что это вычисление дает ответ "да". Они совместно пересчитывают значения двух числовых переменных l и r . В начальный момент $l = 0$, $r = 2^{q(|x|)}$, $C_l := C(0)$, $C_r :=$ стандартная заключительная конфигурация с ответом "да".

- Ход (М) — пара (C_t, t) , причем он утверждает, что (М.1): $t = (r+l)/2$ (легко проверить) и что (М.2): $C_t = C(t)$, чего (А) непосредственно проверить не может.
- Ход (А) состоит в выборе варианта пересчета: $l := t$ или $r := t$ (выбирается одна из половин отрезка).
- Игра останавливается после $q(|x|)$ пар шагов. При честной игре в конце получается $r = l + 1$.
- Предикат выигрыша: Когда одного из игроков удастся уличить в обмане (если нарушено (М.1) или допущена ошибка в пересчете l, r), то нарушивший первым — проиграл. Если то, что (М) выдает в конце игры за $C(l)$ и $C(l + 1)$ (т.е. C_l и C_r) в самом деле не есть 2 последовательные конфигурации машины Тьюринга M , то выиграл (А). В остальных случаях выиграл (М).

Докажем, что эта игра в самом деле распознает язык L . В случае $x \in L$ выигрыш (М) обеспечен тривиальной стратегией — ему надо просто играть честно.

Разберем случай $x \notin L$. Тогда "правильной" допускающей последовательности конфигураций не существует и (М) вынужден обманывать.

До первого хода уже определены две пары (C_l, l) и (C_r, r) : с $l = 0$, $r = 2^{q(|x|)}$; конфигурация C_l — известная всем начальная, а C_r — та допускающая заключительная конфигурация, возможность получения которой в результате вычисления $M(x)$ утверждает (М). При этом на самом деле выполнено условие

$$C_l = C(l) \wedge C_r \neq C(r).$$

Если (М) не нарушит правила, то (А) своим ходом может добиться сохранения этого условия:

1. Если (M) нарушит правила, т.е. $t \neq (l + r)/2$, то (A) уже выиграл.
2. Пусть (M) выберет C_t так, что за $t - l$ шагов машина M не переводит C_l в C_t . Тогда (A) следует выбрать левую половину, т.е. присваивание $r := t$.
3. Пусть (M) выберет C_t так, что за $t - l$ шагов машина M переводит C_l в C_t . Тогда (A) следует выбрать правую половину, т.е. присваивание $l := t$.

Если досрочного выигрыша не произошло, то условие справедливо и в конечный момент, когда $r = l + 1$. По правилам игры (A) в этом случае также выигрывает. ■

Глава 2

Полные задачи для класса $PSPACE$ и классов полиномиальной иерархии.

Легко видеть, что класс $PSPACE$ и каждый класс полиномиальной иерархии “замкнут вниз” относительно сводимости \leq_m^p : если \mathcal{C} – один из них, то для любых двух языков L_1, L_2 выполняется условие

$$L_1 \leq_m^p L_2, L_2 \in \mathcal{C} \Rightarrow L_1 \in \mathcal{C}.$$

По аналогии с понятием NP -полных задач определим Σ_n^p - и Π_n^p -полные задачи как наибольшие (по отношению \leq_m^p) элементы соответствующих классов. Ниже мы приводим типовые примеры таких задач.

2.1 Квантифицированные булевы формулы

Определение 2.1 *Квантифицированной булевой формулой* называется формула вида

$$Q_1 x_1 \dots Q_k x_k \varphi, \tag{2.1}$$

где φ – булева формула, $Q_i x_i, i = 1, \dots, k$ – кванторы существования или всеобщности по булевым переменным x_i . Квантифицированная формула называется Σ_n -формулой (Π_n -формулой), если она начинается с квантора \exists (соответственно \forall) и имеет $n - 1$ чередований кванторов. Квантифицированная булева формула (2.1) *замкнута*, если все переменные ее бескванторной части φ лежат среди x_1, \dots, x_n .

Примеры.

- Формула $\forall x \exists y (x \vee y)$ является истинной замкнутой квантифицированной булевой формулой.
- Формула $\exists x (x \wedge \neg x)$ является ложной замкнутой квантифицированной булевой формулой.
- Формула $\exists x (x \wedge \neg y)$ является незамкнутой квантифицированной булевой формулой, истинностное значение которой зависит от истинностного значения переменной y .

Заметим, что

$$\exists x \varphi(x) \Leftrightarrow \varphi(0) \vee \varphi(1), \quad \forall x \varphi(x) \Leftrightarrow \varphi(0) \wedge \varphi(1),$$

поэтому введение булевых кванторов в пропозициональный язык не изменяет выразительных возможностей, но позволяет записывать некоторые утверждения существенно короче.

Лемма 2.2 *Каждая булева функция $f(x_1, \dots, x_n)$ схемной сложности $c(f)$ может быть (за полиномиальное время) представлена квантифицированной булевой формулой длины $O(c(f))$ в каждом из видов Σ_1 и Π_1 .*

Доказательство. Пусть булева схема для вычисления f размера $c(f) = k + 1$ есть

```

z1 <- t1;
...
zk <- tk;
y <- t.

```

Тогда представляющие f формулы таковы:

$$\begin{aligned} \Sigma_1\text{-вид: } & \exists z_1 \dots \exists z_k ((z_1 \leftrightarrow t_1) \wedge \dots \wedge (z_k \leftrightarrow t_k) \wedge t), \\ \Pi_1\text{-вид: } & \forall z_1 \dots \forall z_k ((z_1 \leftrightarrow t_1) \wedge \dots \wedge (z_k \leftrightarrow t_k) \rightarrow t). \end{aligned}$$

■

2.2 Полные задачи для классов полиномиальной иерархии.

Теорема 2.3 Пусть $n \geq 1$. Задача распознавания истинности замкнутых квантифицированных булевых Σ_n -формул Σ_n^P -полна. Аналогичная задача для Π_n -формул Π_n^P -полна.

Замечание. Мы предполагаем фиксированным побуквенное кодирование формул словами в алфавите $\{0, 1\}$. Задачи распознавания истинности формул указанных видов формализуются как задачи распознавания языков, состоящих из кодов формул. Двоичный код формулы F ниже обозначается через “ F ”.

Доказательство. Сначала проверим, что эти задачи лежат в соответствующих классах. Пусть дана квантифицированная булева формула F вида (2.1). Заменяем в ней каждый блок одноименных булевых кванторов на один квантор по двоичным словам длины m , равной максимуму количества кванторов в таких блоках. Например,

$$F = \underbrace{\forall x_1 \forall x_2}_{\text{блок}} \underbrace{\exists y_1}_{\text{блок}} \underbrace{\forall z_1 \forall z_2}_{\text{блок}} \varphi(x_1, x_2, y_1, z_1, z_2)$$

преобразуется в

$$F' = \forall x_{|x|=2} \exists y_{|y|=2} \forall z_{|z|=2} \varphi(\pi_1(x), \pi_2(x), \pi_1(y), \pi_1(z), \pi_2(z)),$$

где π_i – функции “ i -тый бит слова”. Теперь заметим, что истинностное значение выражения под кванторами

- (а) не изменится при удлинении слов x, y, z дописыванием справа произвольных битов до длины $n = |F|$,
- (б) может быть вычислено за полиномиальное время по формуле F и словам x, y, z , т.е.

$$(F - \text{истинна}) \Leftrightarrow \forall^p x \exists^p y \forall^p z R(“F”, x, y, z),$$

где полином $p(n) = n$, а R – соответствующий предикат из класса P .

Разумеется, то же самое проходит с любой квантифицированной формулой. При этом если F есть Σ_n (Π_n) -формула, то соответствующая правая часть будет Σ_n^P (Π_n^P) -предикатом.

Теперь покажем, что произвольный язык $L \in \Sigma_n^p$ будет \leq_m^p -сводиться к задаче распознавания истинности квантифицированных булевых Σ_n -формул. (Случай с Π_n^p полностью аналогичен.) Пусть

$$x \in L \Leftrightarrow \exists^p w \forall^p b \dots R(x, w, b, \dots), \quad (2.2)$$

где p - полином, а $R \in P$. По доказанной ранее теореме ?? $P \subset P/Poly$, откуда схемная сложность булевой функции f , вычисляющей истинностное значение предиката $R(x, w, b, \dots)$ по битам его аргументов, есть функция полиномиального роста. Более того, доказательство теоремы ?? предлагало прямую (полиномиальную по времени) конструкцию построения соответствующей схемы S_f по числу $m = |x| + |w| + |b| + \dots$, которое, в свою очередь, легко восстановить по $|x|$.

Применим лемму 2.2 к схеме S_f и получим представление булевой функции f квантифицированной булевой Σ_1 или Π_1 -формулой F_f . При этом выберем то из них, в котором кванторы совпадают с самым правым квантором в (2.2). Тогда

$$x \in L \Leftrightarrow \underbrace{\exists w^1 \dots \exists w^{p(|x|)} \forall b^1 \dots \forall b^{p(|x|)} \dots F_f}_F,$$

где формула F справа есть квантифицированная булева Σ_n -формула (новых чередований кванторов не добавилось), а отображение

$$x \mapsto |x| \mapsto m \mapsto S_f \mapsto F_f \mapsto F$$

(т.е. сводящая функция) вычислимо за полиномиальное время. ■

2.3 Пример $PSPACE$ -полной задачи.

Класс $PSPACE$ также содержит наибольшие элементы – $PSPACE$ -полные задачи. К ним относится задача распознавания истинности замкнутых квантифицированных булевых формул. Она представлена языком TQBF, состоящим из (кодов) всех истинных замкнутых квантифицированных булевых формул

$$Q_1 x_1 \dots Q_n x_n \varphi(x_1, \dots, x_n),$$

($Q_i x_i$ – квантор \forall или \exists по булевой переменной x_i).

Лемма 2.4 $TQBF \in PSPACE$.

Доказательство. Определение истинности квантифицированной формулы F в предваренном виде легко представить игрой. Добавлением фиктивного квантора \exists слева можно свести общий случай к

$$F = \exists x_1^1 \dots \exists x_1^k \forall y_1^1 \dots \forall y_1^l \exists \dots \varphi.$$

(M) играет за квантор существования, (A) – за квантор всеобщности. Они оба делают ходы длины $|F|$:

$$\begin{aligned} w_1 &= w_1^1 w_1^2 \dots \\ b_1 &= b_1^1 b_1^2 \dots \\ w_2 &= w_2^1 w_2^2 \dots \\ &\dots \end{aligned}$$

Количество ходов можно также взять $|F|$; оно мажорирует число чередований кванторов в формуле. Предикат выигрыша есть

$$R(F, w_1, b_1, w_2, \dots) \Leftrightarrow \varphi[w_1^1/x_1^1, \dots, b_1^1/y_1^1, \dots]$$

(вместо переменных подставляются первые биты соответствующих ходов игроков). Истинность F эквивалентна условию существования выигрышной стратегии для (M). ■

Теорема 2.5 Язык TQBF является PSPACE-полным.

Доказательство. Докажем, что произвольный язык $L \in PSPACE \leq_m^P$ сводится к TQBF. Рассмотрим игру с полиномиальным числом ходов, распознающую L :

$$x \in L \Leftrightarrow \underbrace{\exists^p w_1 \forall^q b_1 \dots}_{q(|x|)} R(x, w_1 b_1 \dots),$$

где p, q – полиномы, а $R \in P$. Заменим кванторы по словам на блоки кванторов по булевым переменным, а предикат R – на вычисляющую его булеву схему полиномиального размера:

$$x \in L \Leftrightarrow \underbrace{\exists w_1^1 \dots \exists w_1^{p(|x|)} \forall b_1^1 \dots \forall b_1^{q(|x|)} \dots}_{p(|x|) \cdot q(|x|)} (S(x, w_1, b_1 \dots) = 1).$$

Условие $S(x, w_1, b_1 \dots) = 1$ заменим эквивалентной квантифицированной булевой Σ_1 -формулой G полиномиальной длины (по лемме 2.2). Получим квантифицированную булеву формулу

$$F(x) = \exists w_1^1 \dots \exists w_1^{p(|x|)} \forall b_1^1 \dots \forall b_1^{p(|x|)} \dots G,$$

для которой

$$v \in L \Leftrightarrow F(v) \in \text{TQBF}.$$

Все преобразования $v \mapsto F(v)$ сами могут быть проведены за полиномиальное время, что и доказывает сводимость $L \leq_m^p PSPACE$. ■

Литература

- [1] А. Ахо, Дж. Хопкрофт, Дж. Ульман Построение и анализ вычислительных алгоритмов. М.: Мир, 1979.
- [2] М. Гэри, Д. Джонсон Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.
- [3] А. Китаев, А. Шень, М. Вялый. Классические и квантовые вычисления. М.: МЦНМО, ЧеРо, 1999.
- [4] Дж. Сэвидж. Сложность вычислений. М.: Изд-во “Факториал”, 1998.
- [5] P. Gacs, L. Lovasz. Complexity of Algorithms. 1999. <http://www.cs.yale.edu/HTML/YALE/CS/nyPlans/lovasz/complex.ps>
- [6] J. van Leeuwen, ed. Handbook of Theoretical Computer Science. Volume A. Algorithms and Complexity. Amsterdam et al.: Elsevier / Cambridge, MA: MIT Press, 1990.
- [7] M. Sipser. Introduction to the Theory of Computation. Boston: PWS Publishing Company, 1997.